

B A L L I S T I C   R E S E A R C H   L A B O R A T O R I E S

TECHNICAL NOTE NO. 104

September 1949

PREPARATION OF PROBLEMS FOR THE BRL CALCULATING MACHINES

Approved for public release -  
Distribution is Unlimited

---

A B E R D E E N   P R O V I N G   G R O U N D  
M A R Y L A N D

TABLE OF CONTENTS

	Page
I      Introduction . . . . .	3
II     Organization of a Representative Digital Calculating Machine. . . . .	5
III    Criteria of Suitability. . . . .	9
IV    The Eniac. . . . .	11
V    The Bell Relay Computer. . . . .	15
VI   The IBM Relay Calculators. . . . .	17
VII   Application of Criteria of Suitability. . . . .	21
VIII   Information Required by Coder. . . . .	23
IX   Numerical Example. . . . .	25
X   The Differential Analyzer. . . . .	35

**PREPARATION OF PROBLEMS FOR THE BRL CALCULATING MACHINES**

Project TB3-0007

**I-INTRODUCTION**

The Ballistic Research Laboratories have in operation at present four large scale calculating machines: THE ENIAC, THE BELL RELAY COMPUTER, THE IBM RELAY CALCULATORS and THE DIFFERENTIAL ANALYZER. The first three are digital machines. The last is an analogue machine. This report is prepared for the information of persons proposing problems for solution by these machines. It treats the following two related topics:

- (a) determination of the general suitability for machine execution of a proposed method of solution.
- (b) general considerations on the preparation of the method for coding.

The machine computing facilities of the Ballistic Research Laboratories are subject to frequent changes, due both to the addition of new machines and the modification of existing machines. Further, new techniques for using the existing machines are constantly being developed, even without changes in their physical construction. This note represents a first attempt to formulate in general terms the composite ideas concerning problem suitability and preparation of personnel engaged in operating and preparing problems for the currently used BRL machines in their current condition. As such, the information which it contains must be considered as tentative. When the BRL computing facilities change appreciably, or when additional operating experience leads to a further crystallization of ideas, this note will be revised accordingly.

Our discussion of suitability will be limited to the presentation of a set of conditions designed to help answer the following two questions. First, is it physically possible to fit the proposed method onto the machine? Second, if so, what will be the expense in terms of machine and personnel time of obtaining the required results? We do not attempt to discuss any of the other considerations upon which final decision as to the acceptance or rejection of a problem is based.

The second topic, "preparation of the method for coding" is included to give the problem preparer some idea of the type of detailed information which he will be called upon to supply. All of the preparation and coding must be performed prior to putting the problem on the machine. Extensive changes after the problem is on the machine are expensive.

Sections II - IX of this report pertain to the three digital calculating machines. The differential analyzer is treated separately in Section X.

Descriptions of the machines are limited here to operational information necessary to appreciate the criteria of suitability and the details of problem preparation discussed. Readers interested in the preparation of problems for a particular one of the three digital machines may omit the sections on the other two digital machines and the section of the differential analyzer without disturbing the continuity of the report. Readers interested in the differential analyzer only may omit sections II - IX. More complete descriptions of the four machines are given in references (1) - (8). This list of references does not purport to be complete.

#### REFERENCES

- (1) The ENIAC, An Electronic Computing Machine by D. R. Hartree, Nature Magazine, October 12, 1946.
- (2) Description of the ENIAC and Comments on Electronic Digital Computing Machines by Eckert, Mauchley, Goldstine and Brainerd. (A report of the Applied Mathematics Panel of the National Defense Research Committee)
- (3) A Bell Telephone Laboratories Computing Machine by Franz L. Alt, MTAC, V3, No. 21 of Jan. 1948, p. 1f and MTAC V3, No. 22 of April 1948, p. 69f.
- (4) The IBM Pluggable Sequence Relay Calculator by W. J. Eckert, MTAC V3 No. 23, July 1948.
- (5) Instructions for Operating the IBM Relay Calculators by F. L. Alt, Nov. 1945 (Unpublished).
- (6) Programming Principles for the IBM Relay Calculators by J. Lynch and C. E. Johnson, BRL Report No. 705.
- (7) The Differential Analyzer by V. Bush, Journal of the Franklin Institute 212, 1931.
- (8) Report on the Differential Analyzer by A. A. Bennett, ERL Report No. 319, 1942.

## II-ORGANIZATION OF A REPRESENTATIVE DIGITAL CALCULATING MACHINE

The three digital machines have very similar logical organizations and possess a number of features in common. To avoid unnecessary repetition, we discuss the logical organization of a representative digital calculating machine, where the term "representative digital calculating machine" will be understood to refer to any one of the three Ballistic Research Laboratories machines considered here.

A simplified block diagram of such a machine is given in Figure 1.

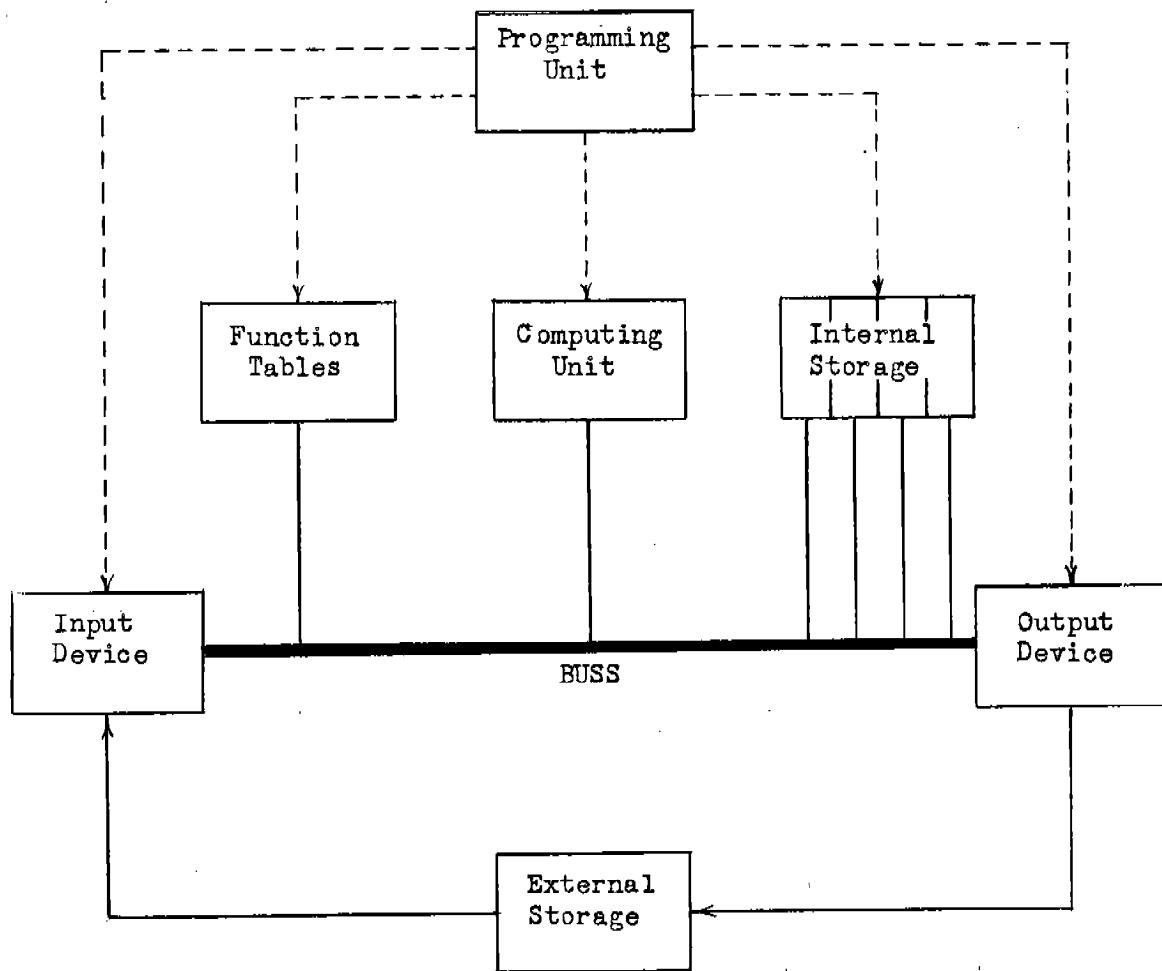


Figure 1 - Block Diagram of the Representative Digital Calculator

The various blocks in the diagram represent the principal components of the calculator. A finite sequence of decimal digits together with an algebraic sign and a decimal point will be called a "quantity." The solid lines, including the heavy horizontal line labeled "buss", represent paths over which quantities may be routed through the calculator. The dotted lines indicate the control exercised by the programming mechanism over the machine's other components. The following statements will serve to explain the diagram and to define some of the terms employed.

The computing unit is a component of the machine capable of performing the arithmetic operations of  $+$ ,  $-$ ,  $\times$ ,  $\div$ ,  $\sqrt{\quad}$ . These operations will hereafter be referred to as "the elementary arithmetic operations." The elementary arithmetic operations are automatic in the sense that the programming necessary to execute them is permanently wired into the machine. In multiplication, for example, it is not necessary for the coder to program each separate addition and shifting operation of which that process consists, but only to give a single order saying "multiply." The internal permanent programming of the computer takes over from there. The computing unit is designed to operate at a speed which is very great compared to the speed of manual computation.

In order to introduce quantities to the computing unit at a speed consistent with its speed of operation, an input device is provided. The problem data is placed manually into the input device at the beginning of the computation. The input device is automatic in the sense that its data is fed out to the computing unit (or some other component of the machine) during the computation as and when required by the machine, without the intervention of an operator.

Similarly, an output device is provided by means of which the machine automatically records its final results.

Suppose that it is desired to evaluate the expression  $(a + b)/(c - d)$  where the quantities  $a$ ,  $b$ ,  $c$  and  $d$  are four ~~entries~~ placed in order in the input device.  $a$  and  $b$  are read into the computing unit and that unit is directed by the programming unit to add these quantities. Some means of holding the quantity  $a + b$  is required now while the computing unit is busy evaluating  $c - d$ . The internal storage of the machine satisfies this requirement. The internal storage is then a component of the machine which will temporarily store for later use intermediate quantities evaluated during the course of a computation. It consists of a number of compartments, in each of which a quantity may be stored. These compartments are called storage registers, or accumulators, depending upon the calculator in which they occur. When referring to the representative digital calculator, they will be called storage registers. Quantities may be directed into and out of the storage registers automatically under control of the programming unit.

If a problem is sufficiently involved, a large number of quantities will need to be stored; so many, in fact, that it is impossible to provide internally all of the storage capacity which may be required. The evaluation of the solutions of a partial differential equation by a stepwise procedure may, for example, require the simultaneous storage of many dif-

ferent quantities at each point of some region of a grid. In cases where the requirements exceed the internal storage capacity available, one follows the obvious procedure of directing the intermediate quantities computed to the output device to be fed back into the machine later, via the input device. This procedure is known as employing external storage. In Figure 1, then, the block labeled "external storage" represents paper tapes, punched cards, or some other similar medium, upon which intermediate results are recorded.

The internal and external storage furnish the calculator with the analogues of the mathematical symbols of ( ), [ ], { }, and other designations of association. In general, the internal storage provides for the temporary storage of a limited number of intermediate quantities in such a manner that they are available to the other components of the machine at a speed consistent with the speed of operation of the computing unit. The external storage sacrifices the availability of the quantities stored in the interests of providing a larger capacity for storage. In comparing machine computing with manual computing, we may liken the internal storage and external storage respectively to the memory of the computer, and the data sheet upon which he enters intermediate results.

Manual computers have frequent occasion to refer to tables of pre-computed or empirical functions. To perform the analogous operation upon the representative digital calculating machine, function tables are provided. A function table is a component upon which entries and interpolation coefficients of a tabulated function can be set up manually in advance. An entry and the associated interpolation coefficients corresponding to an arbitrary argument may then be read out to some other component of the machine automatically when required during the computation.

A function table differs in principle from an input device in the manner in which the various entries are identified. The order in which quantities are to be read from an input device are assumed known in advance. Therefore, at the beginning of a problem, the input data can be arranged manually in the order in which the various quantities will be called for. But the order in which the entries of a tabulated function will be called for is not known in advance. Each entry required must, instead, be identified by an argument which generally arises during the course of the computation. For this reason, means are provided for indexing a function table in such a way that, once the argument is given, the calculator will automatically select the entries associated with that argument. In practice, this distinction between the handling of a set of input data and a tabulated function is an over simplification, since much of the same equipment is used in each case.

In order to transfer quantities from one part of the calculator to another, the various components of the machine are connected by means of a multi-conductor path known as a buss or digit line or multiple. The transfer of a quantity from one component to another is effected by requiring the former component to emit its quantity to the buss, and the latter component to pick up the quantity appearing on the buss simultaneously.

Finally, the machine is equipped with a program control component which directs the other components of the machine as to which of their operations they should perform, and when they should perform them. The program control component is such that the directions to the machine are set up upon it manually in advance in the form of combinations of standard orders. When the calculation is started, the machine executes these orders in sequence, except when directed by a discrimination order to do otherwise. In the example of the evaluation of  $(a+b)/(c-d)$  referred to above, the coder sets up three orders as follows:

- (a) Read the first two quantities from the input device to the computing unit, add them, and transfer the result to Storage Register 1.
- (b) Read the next two quantities from the input device to the computing unit (the first two having now been disposed of), subtract them, and transfer the result to Storage Register 2.
- (c) Read the quantity in Storage Register 1 and the quantity in Storage Register 2 to the computing unit, divide them and transfer the result to the output device.

The discrimination order furnishes a means of directing the machine to decide for itself which of two alternate sequences of computations to perform depending upon the outcome of intermediate results. The machine must be given instructions for each of the alternate sequences of computations and must be furnished with a criterion upon which to base its decision. This criterion must be such that the machine can choose between the alternative sequences according to whether an algebraic sign arising at some specified point in the computation is + or -.

The process of translating the computing formulae to a series of coded orders intelligible to the machine and of transcribing these orders to a medium which may be set-up the program control mechanism is called coding.

### III-CRITERIA OF SUITABILITY

From the above description of a representative digital calculating machine, we may infer certain necessary conditions which a given method of solution must satisfy in order to be suitable for digital machine execution.

Before stating these conditions, we pause to point out explicitly the fact that the machine does not solve a problem in the sense in which the term "solve" is usually understood. The machine will execute a sequence of arithmetic operations designated by the problem preparer. It will select one of several alternate sequences of operations to perform provided that it is given a criterion upon which to base its selection. The machine will not invent the sequence of operations necessary to solve the problem. Thus the method of solution must be devised by the problem preparer. The machine will only execute the computations required by this method. The conditions of suitability to which we will refer are conditions upon the method of solution devised by the problem preparer, not conditions upon the problem itself.

A first necessary condition of suitability which a method must satisfy is that the expressions to be evaluated in its execution are expressible to a sufficient degree of approximation, in terms of the elementary operations and pretabulated functions. A method which satisfies this condition will be said to be reducible.

A second condition which must be imposed upon a method for solution is that it be containable. By this is meant that the capacity of the machine must not be exceeded by the requirements of the method. We use the term capacity in three distinct senses as described below.

In the three operating digital BRL machines, a quantity is represented by a finite sequence of decimal digits, an algebraic sign, and a decimal point (either explicitly represented or understood). By digital capacity is meant the maximum number of decimal digits which may be used to represent a quantity. The digital capacity varies with the machine considered and to some extent varies within an individual machine. The problem preparer must know the digital capacity of the machine with which he is working and must insure that his method is such that this digital capacity will be sufficient to insure the required accuracy of final results.

By storage capacity is meant the number of such quantities which may be simultaneously stored for later use. Internal and External storage capacities are again functions of the individual machine.

Sequencing capacity refers to the numbers of various types of orders which may simultaneously be set up upon the programming unit. The problem preparer will be concerned primarily with the total number of orders which may be used and the maximum number of discrimination orders which may be included among them. The sequencing capacity again depends upon the individual machine. In order for a method to be suitable for machine execution, the digital capacity, storage capacity and sequencing capacity must not be exceeded.

A third condition for suitability is that the method be economical. To appreciate this feature, it is necessary to have some understanding of the steps to be taken to obtain a solution subsequent to the invention of the method of solution, and to have a rough estimate of the expense of these various steps. First, the problem must be coded. This involves only the efforts of the coder. It does not tie up the machine or personnel concerned strictly with operation. Second, the machine must be physically prepared for the problem and the coding must be operationally tested. This involves both machine time and the coder's time. Third, the problem must be "solved" by the machine. This requires running time. The machine, of course, is tied up, but the services of the coder are no longer necessary.

Economy of the coder's time requires that the method of solution be repetitive - repetitive in the sense that the same sequence of operations be performed repeatedly, each time upon different numerical quantities. If the method does not satisfy this condition, the expenditure of manual effort by the coder outweighs the saving of effort achieved by automatic execution of the plan of solution. For any such problem, hand computing is unquestionably cheaper.

Economy in set-up requires that a sufficiently large portion of the problem be "solved" by one set-up to justify the manual effort involved in making that set-up. Specifically, this means that it is uneconomical to solve problems involving small quantities of data which are supplied at infrequent intervals and must be processed as soon as they are received, except when the set-up time is small by comparison with the time required to process the data by other methods.

Economy of machine time requires that very lengthy problems be examined to insure that the machine time consumed is warranted by the value of the results obtained, and that, the desired results will be obtained before they have ceased to be of value.

We have given three general conditions involved in determination of the suitability of a method for machine solution; namely, reducibility, containability and economy. These conditions have been discussed only in qualitative terms. To consider them quantitatively, we need more specific information concerning the capabilities of the three digital machines. This information is best obtained by individual consideration of the machines.

#### IV-THE ENIAC

The ENIAC is unique among the three operating BRL digital machines in that its computing, control and internal storage circuits are completely electronic. This feature gives rise to operating speeds many times greater than those of either of the two relay machines.

The ENIAC normally handles quantities consisting of ten decimal digits and algebraic sign. The location of the decimal point for each quantity computed is completely arbitrary until assigned by the coder. It then remains fixed throughout the problem and cannot be changed without altering the coding. In connection with the ENIAC, the term "quantity" will hereafter refer to ten decimal digits with algebraic sign and arbitrarily located decimal point.

The internal storage of the machine consists of 20 accumulators each of which is capable of storing one quantity. Six of the accumulators have some function other than memory and cannot be used for long term storage.

Associated with the ENIAC are three function tables. Each function table consists of 104 "lines", together with related control equipment. A "line" consists of 14 dial switches - twelve 10-place switches and two 2-place switches. A "line" of information is identified by the computer by a 3 digit argument which generally arises during the course of the computation. Both instructional and numerical information may be set up upon the function table switches. The totality of 312 lines may be divided in any manner desired among tabulated functions, input quantities and orders.

In setting up a pretabulated function upon a function table, the following considerations are to be noted.

The argument of the function must be 3 decimal digits, (where the first digit generally specifies which function table and the last two the line number). Since computation is cheap and function table space is limited, it is generally best to tabulate only the ordinates of the function. Differences, if required, can be computed by the machine itself when needed during the course of the computation.

When entries can be restricted to six digits and algebraic sign, then two entries can be set up upon each line, otherwise only one entry per line may be used. Interpolation with continuously varying intervals is, in general, impractical. However, it is feasible to use different tabular intervals within different zones of the tabulated function.

The ENIAC receives its instructions from orders which are set up upon the switches of the function tables. The totality of possible orders which may be employed is known as the "code". The code provides orders for performing the elementary arithmetic operations upon quantities specified by their location in the machine, for controlling the input and

output devices, for looking up entries of tabulated functions, for transferring quantities among the various components of the machine, for shifting decimal points, and for discriminating upon any quantity, either given or computed.

Input quantities other than the entries of a pretabulated function may be introduced to the machine in three ways - from an IBM card reader, from two manually set registers of dial switches, each of which will hold one quantity, or from the function tables. The IBM card reader and the two registers of dial switches are associated with a relay and switch storage unit called the constant transmitter. Upon signal from the programming mechanism, the ENIAC will read 8 quantities from a card and store them in the relays of the constant transmitter. Once a card has been read (a mechanical and therefore relatively slow operation - about .6 seconds), its information is available to the other components of the computer at electronic speeds. The constant transmitter retains the quantities from the card until they are no longer needed, at which time another card may be called for. The two quantities set up upon the registers of dial switches are retained by the constant transmitter until the switches are manually reset by the operator which clearly cannot be done while the machine is computing. Input quantities may be set up upon the function tables in the same manner as upon the switch registers of the constant transmitter. The difference is that the function table quantities are called for by variable arguments, whereas the switch registers must be called for by fixed register identifying symbols.

In general, the two switch registers are used for input quantities which remain absolutely constant throughout a computation. The card readers and function tables may also be used for this purpose when their full capacities are not otherwise utilized. Quantities which change from cycle to cycle of a computation, such as for example datum numbers, are invariably introduced through the card feeds. Quantities which must be identified by a variable argument may be introduced only through the function tables.

All output quantities are recorded upon IBM cards. The card punch and its associated control equipment are known as the printer. The coder can program the machine to punch out any desired quantities. Punching is again a mechanical process requiring about .6 seconds. To utilize fully the enormous electronic speed of operation of the ENIAC, card punching must be held to a minimum. Output quantities recorded upon cards may later be printed as a separate operation by an IBM tabulator.

No special provision is made for external storage. The existent card punch and card feed are used for this purpose. The operator manually transfers the cards from the punch to feed. This is entirely satisfactory in cases where the transfer of externally stored quantities from punch to feed is required only at infrequent intervals - i.e., when relatively long term storage is needed. Short term external storage cannot be used without sacrificing some of the electronic speed of the ENIAC.

The capacity for orders, capacity for entries of tabulated functions and capacity for constants are interrelated. Constants can, and usually are, introduced by the constant transmitter. This leaves the 312 lines of the function tables as the limiting value for the sum of the number of orders and the number of functional entries which may be used in a single set-up. There is no limitation upon the proportion of discrimination orders which may be included in the total number of orders used.

## V-THE BELL RELAY COMPUTER

The Bell Relay Computer employs standard telephone relays for computational and internal storage purposes. Its input, output and program control devices consist of unmodified printing telegraph equipment. The Computer is completely self-checking throughout. It is designed to run unattended for long periods of time. The BRL installation consists of two complete identical computing systems of the type described below. They are operated simultaneously on different problems. The two systems are not used to check each other.

The digital capacity of the Bell Relay Computer is seven significant figures. All quantities handled by the machine are in the standard form  $\pm . yxxxxxx \cdot 10^{\pm ix}$  where y is an arbitrary non-zero decimal digit, the x's are arbitrary decimal digits and i is an arbitrary binary digit. Thus the machine handles all seven significant digit quantities whose absolute values lie between  $+.1000000 \times 10^{-19}$  and  $+.9999999 \times 10^{+19}$  inclusive. The interval  $[-.9999999 \times 10^{-19}, +.9999999 \times 10^{+19}]$  will be called the "range" of the machine.

The computing unit performs the elementary operations upon quantities in standard form, and automatically normalizes the result of any elementary operation to standard form. That is to say, the computer employs a floating decimal point. The result of any elementary operation is automatically rounded to the nearest 7th significant digit. No provision is made for retaining the less significant figures of a  $7 \times 7$  multiplication nor for storing the remainder after a seven significant digit quotient. Thus calculations requiring more than seven significant digits are not feasible upon this machine.

Built into the machine as permanently wired connections are tables of the functions  $\sin x$ ,  $\text{arc tan } x$ ,  $\log x$  and  $10^x$ . For the purposes of the problem preparer, these functions may be considered as tabulated over the entire interval for which both the argument and the functional value are within the range of the machine. The unit of angle is  $90^\circ$ . Entries from the permanent tables are made available to the other components of the machine by switching operations and are thus obtainable without appreciable time delay.

In addition to the permanently built in function tables, any single entry tabulated function may be specially prepared for use with a specific problem. Double entry tables are also possible, but are in general difficult to handle. The arguments and entries of special tables are coded on paper tapes and read to the other components of the machine under control of the programming unit. Complete automatic control features for finding the entry associated with an arbitrary argument are built in. The machine locates the desired argument of a special table by scanning the arguments appearing on the tape upon which the table is coded. This is a relatively slow operation. The form of the table is subject to the following limitations:

- (a) Arguments must lie within the range 0000-9999. They need not be equally spaced.

(b) Entries and interpolational coefficients must lie within the range of the machine.

The internal storage of the machine consists of 15 storage registers and eight sign registers. Each of the storage registers holds a seven significant digit quantity in standard form. Each sign register holds a single algebraic sign to be used for discrimination purposes.

External storage consists of an input mechanism and an output mechanism associated in such a way that the output quantities are automatically transferred to a position where they are available to the input mechanism without the intervention of an operator. The capacity of external storage is unlimited.

The programming unit reads its orders from coded paper tapes. Repetitive sequences of operations are put on closed tapes. Sequences of operations to be performed but once per problem are placed upon two ended tapes. The total capacity for orders is unlimited. About 30 discrimination orders may be included without inconveniencing the coder. Additional discrimination orders may be added ad infinitum with the complexity of the coding increasing rapidly with number of additional discrimination orders included.

Input quantities which remain absolutely constant throughout the course of a problem may be introduced to the machine on the programming tapes. Input quantities which vary with successive iterations of a sequence of operations are introduced on special paper tapes in the same manner as special tabulated functions. No provision is made for introducing input quantities via switch settings as may be done on the ENIAC and the IBM Relay Calculators. All input quantities must be of the standard seven significant digit form.

Output quantities may be punched on paper tape for later transcription to printed form or be printed directly on a typewriter as they are computed.

## VI-THE IBM RELAY CALCULATORS

Among the high speed computing machines presently operating in the Computing Laboratory there are two IBM Relay Calculators. These machines, identical in every respect, were built by the International Business Machines Corporation and installed at Aberdeen Proving Ground, Maryland, early in 1945. They have been in continuous operation ever since, except for occasional short periods of time when the incorporation of certain improvements necessitated their shut-down.

Following is a description of certain features of the machines, a knowledge of which will materially assist anyone who is interested in the preparation of problems to be solved by means of them. The reader who is concerned with a more detailed knowledge of these calculators would do well to consult references (4), (5), (6).

Data to be introduced into the machine must be punched on standard IBM cards of 80 columns, and then read into storage or computing counters by means of the two separate card feeding units, which may read as many as four cards simultaneously, while punching results on a fifth card. This reading and punching action goes on at the rate of 100 cards of 80 columns each a minute. There are, further, 24 dial switches allowing for the introduction into the machine of 24 digits that may be used as constant factors.

All quantities enter the machine in the normal decimal notation; each number may have up to twelve digits with a plus or a minus sign. Since many mathematical functions, such as trigonometric, exponential, logarithmic functions, and many others, are used very frequently on problems put on the machine, they have been punched on cards, and are thus available in permanent form. The list of functions thus available, while too lengthy to be given here, may be had on request. In cases where special functions must be employed they can, if not too extensive, easily be punched. However, to effect an economy in the time needed for table look-up the arguments should best be arranged in monotonic order. While results of calculations obtained on the machine are always punched on cards, they may also be listed on sheets of paper of any desired length, or, in cases where the results can be arranged in tabular form, they can be printed by means of the electro-magnetic typewriter. (For a description of this machine see the article by Eckert and Haupt, The Printing of Math Tables, MTAC Vol. II, No. 17, Jan. 47.)

The machine performs all operations (receiving, transmitting, comparing, shifting, rounding, and recording of data) and calculations (addition, subtraction, multiplication, division, and square rooting) by means of electro-mechanical relays arranged in 36 groups, the "counters".

~~26~~ <sup>Thirty</sup>one of these counters can accommodate numbers with not more than six digits, and their sign, plus or minus; the other five counters can store twelve digit numbers. A negative number is stored in a counter as its complement with respect to 999 999. These groups of relays, in conjunction with the twenty-dial switches mentioned above, permit an internal storage of up to ~~240~~ <sup>240</sup> digits, or ~~640~~ <sup>640</sup> in cases where both calculators are hooked up so as to operate as a single unit.

The machine receives its operational instructions mainly from two removable plug-boards. The three-panel board generally assigns the channels through which the reading and punching of data is controlled, while the two-panel board serves to control the sequencing of operations. These boards have to be wired manually for different types of problems, but, once programmed, the set-up may be stored away in symbolical form for future use on similar problems.

While, as described above, apparently somewhat limited in internal storage capacity the machine is, nevertheless, very efficient in that it has great overall speed of operation, is very flexible in its programming, card reading and recording facilities, and is relatively easy to operate.

The timing of all operations performed within the machine is provided by a shaft in the card reading unit rotating at the rate of 100 rpm. During each revolution of this shaft, while one card is being read or punched, two timing circuits produce impulses that may be used to set relays or initiate the reading or punching of cards. The 48 impulses generated by one of these timing circuits, occurring at 0.0125 second intervals, form the basic sequence of "points" in the timing schedule of the machine. The time required for a relay to function properly is approximately one sequence point.

On certain sequence points in the 48 point cycle - or, if desired, in the 96 point double-cycle represented by two revolutions of the timing shaft - the machine can be made to perform certain operations, which are then repeated during each following cycle. The sequence of operations which the machine carries out is thus periodic, the period being of 48 (or 96) sequence points' duration.

As to elementary operations, it takes two sequence points to add or subtract numbers of as many as 12 digits. Multiplication is limited to factors of six digits; the twelve digits product is available 16 sequence points, i.e., 0.20 seconds, later. However, it is possible to space successive multiplications in such a way as to effect a time saving of 0.05 seconds. Division is limited to a twelve digit numerator and a six digit denominator; the average dividing time is 0.2 seconds per quotient digit, of which there may be twelve. Finally, square rooting is limited to a twelve digit square and a six digit root; it takes about the same amount of time to get a root digit as it does to get a quotient digit.

Numbers read out of certain counters may also be shifted up to six places to the right or to the left. Rounding of positive numbers is easily achieved by simply adding a five into the decimal place to be dropped. Finally, two numbers may be compared, and the machine programmed to carry out alternate sequences depending on the result of this comparison.

Since the machine can perform only the elementary mathematical operations cited above it is thus obviously required to break down all necessary mathematical manipulations - with the exception of functions punched on cards in permanent form - into combinations of finite numbers of elementary operations.

While the Relay Calculators can do certain types of jobs more easily and efficiently than other machines the amount of work involved in the actual preparation and coding of a particular job can be quite considerable. First a time chart of the operations has to be drawn up. Next a plug board diagram has to be made up, and then the actual wiring of the plug board made on the basis of this program.

A great many different types of problems have been solved on these machines. To mention only a few:

Solution of Partial Differential Equations of the Heat Conduction Type by Schmidt's Method.

Solution of Large Systems of Simultaneous Linear Equations by Gauss-Jordan's Method.

Construction of Tables of the Incomplete Beta Function by Recursion Formulas.

The solution of these and many other problems became possible only after due consideration of the criteria stated in section III; to the reader who has a problem of his own, we recommend a careful scrutiny of the pertinent parts of that section.

VTT -- APPLICATION OF CRITERIA OF SUITABILITY

It was pointed out in section III, that in order to be suitable for execution by a digital calculating machine, a method for the solution of a problem must satisfy conditions of reducibility, containability and economy. The discussions of the individual machines given in sections IV, V, and VI are intended to provide the quantitative information necessary to apply these conditions to a particular method of solution proposed for a particular machine. In Table I, we summarize the principal machine characteristics upon which the above enumerated conditions of suitability depend.

TABLE I

DEPENDENCE OF SUITABILITY CONDITIONS  
UPON MACHINE CHARACTERISTICS

Conditions of Suitability	Machine Characteristics		
	Eniac	Bell Relay Computer	IBM Relay Calculators
Reducibility	$+, -, \times, \div, \sqrt{\phantom{x}}$ and pretabulated functions		
Containability			
Digital Capacity	10 digits fixed decimal point	7 digits floating decimal pt.	6 or 12 digits fixed decimal point
Internal Storage	20 quantities  (200 digits)	15 quantities  (105 digits)	26 6-digit quantities 5 12-digit quantities (216 digits)
External Storage	Unlimited (manual access)	Unlimited (automatic access)	Unlimited (manual access)
Programming Capacity	Limited	Unlimited	Limited
Economy	Several      Days		
Coding Time	Several hours	Several hours for 1st set-up zero-time for subsequent set-ups	Several hours
Testing and Set-Up Time			
Running Time	.012 sec. per mult.	1 sec. per mult.	.2 sec. per mult.

Methods suitable for one machine are not necessarily suitable for another. Methods suitable for all machines are frequently more suitable for one machine than for another.

If a method appears to call for several months of manual computation by a large staff of computers, one might start thinking of the Eniac. If a non-repetitive method calls only for several man hours of computation, no machine will be suitable. Methods lying somewhere between these extremes may be suitable for the Bell Relay Computer and the IBM Relay Calculators.

Methods requiring extremely lengthy schedules of operations to be performed may be unsuited for the Eniac and the IBM Relay Calculators due to programming capacity requirements in excess of those available. The Bell Relay Computer is not subject to this type of limitation.

Problems requiring the handling of large quantities of data are best suited to the IBM Relay Calculators provided that other considerations of suitability are equal. Handling of input and output is more bothersome on the Bell Machine. Input and output can be readily handled by the Eniac. However, if extensive input and output is required for a small amount of computing, the electronic speed of the machine is wasted for the most part since the whole problem is slowed down to card handling speeds.

Problems requiring frequent set-up, such as the processing of intermittently received data are best suited to the Bell Relay Computer unless the computation itself is too time consuming. The set-up times of the Eniac and IBM computers, make it uneconomical for those machines to process small quantities of data intermittently.

The intermediate or final results of some computations may vary considerably in magnitude from cycle to cycle. In problems of this nature, the Bell Relay Computer has a distinct advantage over the other two digital machines in that its floating decimal feature obviates the necessity for extensive use of scale factors. It is theoretically possible to program a floating decimal point on the Eniac, but practical considerations make this type of programming rarely, if ever, feasible.

Certain dodges are possible to modify a method of solution so as to make it more adaptable to a particular machine. Since the Eniac has relatively small sequencing and storage capacity but high speed of operation, it is usually desirable to carry out stepwise approximations on this machine by using low order approximations, small intervals and many steps. Since the Bell Machine has unlimited sequencing capacity, large automatic external storage but slow speed of operation, it is generally best for this machine to use high order approximations, large intervals and few steps. The IBM Relay Calculators lie somewhere in between the Eniac and the Bell Relay Computer in this respect.

## VIII-INFORMATION REQUIRED BY CODER

In order to code for machine execution a method for the solution of a problem, the coder must have at hand the specific information listed below:

- (a) A list of formulas expressing all quantities to be computed in terms of the elementary operations (+, -,  $\times$ ,  $\div$ ,  $\sqrt{\phantom{x}}$ ) and pretabulated functions.
- (b) Accuracy required of intermediate and final values.
- (c) An exact criterion for each discrimination required.
- (d) The approximate range of all initial, intermediate and final quantities.
- (e) A list of input quantities, including pretabulated functions, classified according to the frequency with which they change.
- (f) A list of the quantities which the problem preparer desires to have printed.

In addition to the above information required by the coder, it will be necessary that a sample run be computed by other means before set-up and testing can be undertaken.

Collecting these items of information requires a high degree of cooperation between the problem preparer and the coder. If the method to be coded requires subsidiary operations for which standard well known numerical methods are available, (such as, for example, interpolation, numerical differentiation and integration, solution of algebraic equations, evaluation of determinants, etc.), then the selection of that standard method best suited to the machine to be used, and the reduction of that standard method to elementary operations is usually best left to the coder. Suggestions from the problem preparer are always helpful, however.

The accuracy required of intermediate and final values must be specified for several purposes. First, it is needed for all machines in order to determine the number of terms or the number of iterations required of approximate formulas. Second, for the fixed decimal point machines, it is needed in conjunction with information on the range of quantities computed in order to determine suitable scale factors. Third, it is necessary for the IBM Relay Calculators in order to determine whether to employ six or twelve digit quantities in the computation.

Discriminations are used to direct the machine as to which of two alternate sequences of operations to perform. The machine's decision is usually based upon a comparison of two quantities, one or both of which arise during the course of the computation. The quantities to be compared must be specified for each discrimination.

Since the Bell Relay Computer employs a floating decimal point, the only estimation of range required for this machine is the assurance that all quantities lie between  $-.9999999 \times 10^{+19}$  and  $+.9999999 \times 10^{+19}$  inclusive. For the other two machines, a much closer estimate of range is necessary if full digital capacity is to be realized. The decimal point in each register must be located sufficiently far to the right that the leftmost figures of the largest quantity entering the register are not lost. At the same time, the decimal point must be located sufficiently far to the left that the required number of significant digits of the smallest quantity entering the register is retained. If the variable quantity to which that register is assigned changes appreciably in magnitude during the computation, variable scale factors may have to be employed to satisfy both of these conditions. The problem preparer need not list ranges of quantities whose magnitudes are immediately obvious from the computing formulae. For example, if the range of  $x$  is given, the coder can supply scale factors necessary to keep  $1/(1 - x)$  within range without instructions from the problem preparer. But in many cases, the ranges of intermediate and final values cannot be conveniently predicted from the computing formulas alone. In these cases, the ranges should be furnished by the problem preparer. The physical interpretation of the variables in question will frequently help him to determine their range. In cases where approximations are employed, range information is usually available as a by product of the investigation of the order of approximation required.

The method chosen by the coder to introduce input quantities to the machine usually depends upon how frequently these quantities change. For this reason, input quantities should be classified according to this attribute.

Since these machines do not automatically keep a record of intermediate quantities computed, once they have been used, the problem preparer must clearly specify which results he desires to have printed.

The hand computed sample run is required to check the coding and set-up. One application of each formula with a representative set of numerical values should be sufficient. Checking the machine itself is accomplished by other means which need not concern the problem preparer. It is usually desirable to compute the canned run only after the detailed computing formulae have been agreed upon.

## IX-NUMERICAL EXAMPLE

The conditions of suitability and the information necessary for coding will be illustrated by a numerical example. The authors are somewhat embarrassed by the fact that a method simple enough to be analyzed in a few pages is in general so simple as hardly to warrant the use of a large scale machine. Nevertheless, it is believed that such an example will be valuable in clarifying some of the notions previously introduced.

**PROBLEM** ~ It is required to tabulate the family of functions

$$F_a(x) = \int_0^x \frac{\varphi(t) dt}{\sqrt[3]{100 - at}}$$

for the range  $0 \leq x \leq 10$  by intervals of  $\Delta x = .1$  to within an error of  $5 \times 10^{-3}$ , where  $0 \leq a \leq 5$  by  $\Delta a = .1$ , and  $\varphi(t)$  is an empirical function given by the following table:

t	$\varphi(t)$
.0	1.00000
.1	1.00250
.2	1.01070
.	.
10.0	70.00000

We suppose the tabulation of  $\varphi(t)$  to represent a function which is monotonic and continuous and possesses continuous bounded derivatives of all orders upon the interval  $0 \leq t \leq 10$ . Devise a method suitable for machine execution and list all information necessary for coding.

The following method is proposed:

(a) Express  $Z = \sqrt[3]{Y}$ , where  $Y = 100 - at$  by the iterative formula

$$(1) \quad Z_r = \frac{2Y Z_{r-1}^3 + 1}{3Y Z_{r-1}^2} \quad r = 1, 2, 3, \dots$$

We leave the method of selection of the 1st approximation  $Z_0$  and the number of iterations arbitrary for the present.

(b) Represent  $\varphi(t)$  by a family of interpolational polynomials of the  $n$ th degree ( $n$  may be equal to zero), each of which will serve as an approximation to  $\varphi(t)$  on some sub-interval of the range of  $t$ . The degree of the interpolational polynomials and the size of the sub-intervals will be selected later.

(c) Approximate  $F_a(x) = \int_0^x f_a(t) dt$  where  $f_a(t) = \varphi(t)/\sqrt[3]{100 - at}$  by a mechanical quadrature formula expressing  $F_a(x)$  as a linear combination with constant coefficients of selected

equally spaced values of the integrand  $f_a(x)$ .

$$(2) \quad F_a(x) = F_a(x - m\Delta x) + \sum_{j=0}^m C_j f_a(x - j\Delta x)$$

The order,  $m$ , of the quadrature formula will be selected later.

We define a "point" to be the computation of a single value of  $F_a(x)$  for arbitrary  $a$  and arbitrary  $x$ ; a "run" to be the evaluation of  $F_a(x)$  for a single arbitrary value of  $a$  and all required values of  $x$ ; the "problem" to be the evaluation of  $F_a(x)$  for all  $a$  and all  $x$ .

The proposed method is clearly reducible. We will examine its containability and economy.

We first consider containability as regards to sequencing capacity. The following purely arithmetic operations are required:

- (1) 1 division, 1 addition and about 6 multiplications per iteration to extract a reciprocal cube root according to formula (1).
- (2)  $n$  multiplications and  $n$  additions to obtain an interpolated value of  $\phi(t)$ , where  $n$  is the order of interpolation employed.
- (3)  $n$  multiplications and  $n + 1$  additions to evaluate the quadrature formula, where  $n$  is the number of ordinates to be used in a single application of formula (2).

In addition to these arithmetic operations, the machine will have to perform various miscellaneous operations for the following purposes:

- (1) Obtaining a first approximation to the reciprocal cube root.
- (2) Deciding upon the number of iterations required for the reciprocal cube root.
- (3) Selecting from the function table the proper coefficients of the polynomials approximating  $\phi(t)$ .
- (4) Delivering to and extracting from internal storage successive values of the integrand.
- (5) Changing the parameter  $a$  from one run to the next.
- (6) Generating successive values of the independent variable  $x$ .
- (7) Delivering final computed values of  $F_a(t)$  to the output.

Without going into further detail, it may be seen that the performance of the above listed operations are well within the sequencing capacity of all three machines, for any reasonable orders of interpolation and quadrature.

Let us now consider containability as regards to digital capacity. Since  $\varphi(t)$  is continuous, and  $1/\sqrt[3]{100-ax}$  does not change sign on the interval 0 to  $x$ ,

$$F_a(x) = \varphi(t) \int_0^x \frac{dt}{\sqrt[3]{100-ax}} = \frac{3\varphi(t)}{2a} \left\{ 100^{2/3} - (100-ax)^{2/3} \right\}$$

where  $0 \leq t \leq x$ . Furthermore,  $\varphi(t)$  is monotonic and increasing. Therefore,

$$\frac{3\varphi(0)}{2a} \left\{ 100^{2/3} - (100-ax)^{2/3} \right\} \leq F_a(x) \leq \frac{3\varphi(10)}{2a} \left\{ 100^{2/3} - (100-ax)^{2/3} \right\}$$

Now  $\frac{3}{2a} \left\{ 100^{2/3} - (100-ax)^{2/3} \right\}$  increases with both increasing  $a$  and increasing  $x$ . Also  $\varphi(0) = 1$  and  $\varphi(10) = 70$ . Therefore,

$$0 \leq F_a(x) \leq \frac{3 \times 70}{2 \times 5} \left\{ 100^{2/3} - (100-5 \times 10)^{2/3} \right\}$$

or

$$0 \leq F_a(x) \leq 147$$

Clearly, six significant figures are sufficient to represent  $F_a(x)$  to the required accuracy of 3 decimals. Several extra places should be allowed, however, for the accumulation of round off error in adding the quantities  $C_i f_a(t - \frac{i}{5}\Delta t)$ . Thus, the register used to accumulate  $F_a(x)$  should have at least 7, preferably 8, places. This requirement is easily satisfied by the digital capacity of the Eniac. It is barely satisfied by the digital capacity of the Bell Relay Computer. On the IBM Relay Computers, it will be necessary to use a double length (12 - digit) register for the accumulation of  $F_a(x)$ .

The integrand,  $f_a(t) \equiv \varphi(t)/\sqrt[3]{100-ax}$  varies from 0.21 for  $t = 0$  and all  $a$ , to 19 for  $t = 10$ ,  $a = 5$ .  $C_i$  is of the order of  $\Delta x = .1$ . Therefore,  $C_i f_a(t - \frac{i}{5}\Delta t)$  will be expected to be of at most several units in size for all  $a$  and all  $t$ . The quantity  $C_i f_a(t - \frac{i}{5}\Delta t)$  to 5 decimals can therefore be contained in a register of six fixed digits.

Since  $C_i f_a(t - \frac{i}{5}\Delta t)$  is at most several units in size, an accuracy of 6 significant figures will yield the required accuracy of 5 decimals. We should then have at least 6 significant figures of accuracy in each of the factors  $\varphi(t)$  and  $1/\sqrt[3]{100-ax}$ . We can certainly obtain  $\varphi(t)$  from a function table to 6 significant figures without any difficulty as to digital capacity. Also  $1/\sqrt[3]{100-ax}$  may be obtained by repeated application of formula (1) without more than 6 fixed digits in any component quantity of that formula.

Thus the method is containable for all three machines so far as digital capacity is concerned, subject to the following limitations:

- A 12-digit register must be used for accumulating  $F_a(x)$  on the IBM relay calculators.
- There will be some doubt as to the last figure of  $F_a(x)$  for large  $a$  and large  $x$  when computed by the Bell Relay Computer.

Let us now examine the storage capacity required. Clearly the quantities computed every run are independent of those computed on all

other runs, except possibly that the quantity  $a$  may be generated by adding  $\Delta a$  to the value used in the preceding run. At most one quantity need be stored from one run to the next.

Each value of the integral computed requires several values of the integrand. The exact number depends upon the order of the quadrature formula employed. We do not wish to recompute any values of the integrand. We are therefore required to store integrand values from one point to the next. If  $x$  is to be generated from point to point, we shall need a register in which it may be stored. If we are to obtain the first approximation to  $\sqrt[3]{Y}$  from the final approximation of at the previous point, a register will be needed in which this quantity may be stored. To compute  $F_a(x)$  we need also to store from a previous point  $F_a(x-\Delta x)$ . Apparently, something under 10 registers will be sufficient for all storage from point to point.

For temporary storage at a point we need to hold either  $\varphi(t)$  or  $\sqrt[3]{100}$  - at while the other one of these quantities is being computed. We need several working registers for evaluating the interpolation formula for  $\varphi(t)$ . These same working registers may be used for manipulating the components of  $Z_r = (2Y Z_{r-1}^3 + 1) / 3Y Z_{r-1}^2$  and counting the number of iterations required. We can also use these working registers to evaluate the quadrature formula.

It appears that 15 storage registers in all will be sufficient to take care of all variable storage required, both long term and short term. Thus on any of the three machines, the storage can be handled internally. The method is containable from a storage point of view.

Let us now consider some of the economic aspects of the problem. Several days not involving the machine will be required for coding. Several hours of machine time will be needed for set-up and testing. Since the set-up need be performed but once, there is no need to consider subsequent set-up time.

The Eniac will probably require several seconds running time per-point. For 100 points per run and 50 runs, two 24 hour Eniac days should be sufficient to complete the problem, including set-up and testing and allowing for a considerable number of unforeseen delays.

The Bell Relay Computer will probably require several minutes per point or say 150 hours running time for the whole problem. Two weeks around the clock for one of the two duplicate computers should be sufficient to complete the job, including set-up, testing, running and unforeseen delays.

The IBM computers would probably require about one minute per point. With a 40 hour week, two weeks should again be ample for set-up, testing, running and some unpredictable delays.

We will not attempt here to estimate the time required for manual computation of the problem except to point out that in this case, the manual computing time will not be very great compared with the machine running time. This is true for several reasons. First, only 6 digits

accuracy are required of most of the quantities computed. Second, cube roots may be obtained much more simply by hand computer than by machine. In fact, linear interpolation in Barlows' Tables will give more than enough accuracy throughout the whole range of the independent variables. Third, table look-ups in the  $\mathcal{Q}(t)$  table will be extremely simple for the hand computer and can be obtained without interpolation.

On the other hand, the results obtained by machine will be perfectly self consistent with regard to how quantities are rounded, and how much accuracy is retained at each step. This is difficult to accomplish by hand.

Considering all of the above points, we are inclined to believe that the problem, as solved by this method, is suitable for machine execution only in the event that the sufficient number of hand computers necessary to obtain the required numerical results in time are unavailable.

However, for the sake of continuing the discussion, let us suppose that sufficient hand computers are not available and that it has been decided that the problem should be solved by machine. In this event, we would recommend the IBM relay calculators as being most suitable for the problem. The additional flexibility of the Bell machine is unnecessary here, and the additional expense required to operate the Eniac is not justified by the saving of machine time effected.

Before considering information necessary for coding, let us inquire as to the effect on suitability of modifying the problem in various ways.

First, suppose that only 5 or 10 runs are required instead of 50. In this case, running times for all of the machines are proportionately less, but coding, set-up and testing times remain the same. It is now even less economical to execute the solution by machine.

Second, suppose, on the other hand, that two independent variable parameters,  $a$  and  $b$ , enter under the integral sign, and that it is required to make  $50^2 = 2500$  runs instead of only 50 runs. In this case, 50 times as much computing is required. We now have a problem of Eniac dimensions. The problem would be even more suitable for the Eniac if

it were required to tabulate only the function  $\int_0^x \frac{\mathcal{Q}(t) dt}{\sqrt[3]{100 - at}}$  as

a function of  $a$  instead of  $\int_0^x \frac{\mathcal{Q}(t) dt}{\sqrt[3]{100 - at}}$ . In this case, the relatively long time necessary to punch out a single value of the integral (.6 secs) would be required only once per run, instead of once per point.

Fourth, suppose that the function to be tabulated is  $\int_0^x \frac{\mathcal{Q}(10 \sin \frac{a\pi}{t+1}) dt}{\sqrt[3]{100 - at}}$

The argument  $10 \sin \frac{a\pi}{t+1}$  of  $\mathcal{Q}$  no longer increases monotonically with  $t$ . In fact, for small values of  $a$ , it oscillates rapidly. Since the IBM computers can automatically consult a function table with non-monotonic argument, only with the manual intervention of the operator, this introduces a serious difficulty for these machines. In this case,

the only reasonable possibilities are the Eniac or the Bell Relay Computer. On these latter two machines, little difficulty is encountered in handling functions with oscillating arguments.

Finally suppose that  $a$  is allowed to range from 0 to 10 instead of 0 to 5. The integral still exists, though it is improper for  $x = 10$ ,  $a = 10$ . The method, as given, is now unsuited for machine execution (and unsuited for manual execution too) due to the fact that the integrand grows without limit with increasing  $x$  and  $a$ . This difficulty can be removed by employing an expansion of the integrand in powers of  $at$  throughout part of the range, and integrating the first several terms explicitly. Even so, however, the integrands will vary greatly in size with both  $a$  and  $t$ . In this case, the Bell Relay Computer has a distinct advantage over the other two machines in that its floating decimal feature eliminates the necessity for extensive use of scale factors.

On the assumption that the problem is to be run on the IBM computers, let us now proceed to information required for coding. It is first necessary to determine specifically those features of the method which were left arbitrary upon first consideration. Let us decide that for the first approximation to  $Z$  at any point, we will use the final approximation to  $Z$  at the previous point, and that the iteration at a point will be continued until the difference in absolute values of two successive approximations is less than the prescribed tolerance of  $6 \times 10^{-6}$ . Values of the function  $\phi(t)$  are needed for only fifty discrete values of  $t$ . For the IBM computers, it is more advantageous to store all 50 of these values and thus avoid interpolation, than to condense the table and thus save function table space. Let us suppose that the nature of the function  $\phi(t)$  is such that "3-ordinate integration" at the prescribed interval of  $\Delta x = .1$  will give sufficient accuracy. Under these conditions, a list of reduced formulas suitable for coding is given in table II.

We have already considered the ranges of most of the quantities computed, and the maximum allowable errors of the quantities. For the convenience of the coder, the ranges and maximum allowable errors of these and a few other quantities are collected in table III. Input quantities are listed in table IV, and desired output in table V. The first several points of a canned run are given in table VI. Clearly, computation of three values of the integral ( $a = .3$ ,  $x = 0$ ), ( $a = .3$ ,  $x = .1$ ) and ( $a = .3$ ,  $x = .2$ ), require the evaluation of each formula employed at least once, and are thus sufficient for a check on the coding. If, in addition, the problem preparer desires to check his analysis of ranges, errors and formula reductions, he should complete the canned run.

TABLE II - REDUCED FORMULAS

Start with  $a = 0$ , increase  $a$  by steps of  $\Delta a = .1$  through  $a = 5$ .  
 For each value of  $a$ , start with  $\bar{x} = 0$ , increase  $x$  by steps of  $\Delta x = .1$  through  $x = 10$ .

$$F_a(x) = \begin{cases} 0 & \text{for } x = 0 \\ F_a(0) + (\Delta x/12) \{ 5f_a(x-\Delta x) + 8f_a(x) - f_a(x+\Delta x) \} & \text{for } x = \Delta \bar{x} \\ F_a(x-2\Delta x) + (\Delta x/3) \{ f_a(x-2\Delta x) + 4f_a(x-\Delta x) + f_a(x) \} & \text{for } x = 2\Delta x, 3\Delta x, \dots, 100\Delta x \end{cases}$$

where

$$f_a(x) = \begin{cases} 1/\sqrt[3]{100} & \text{for } x = 0 \\ \varphi(x) \cdot Z(x) & \text{for } x = \Delta x, 2\Delta x, \dots, 100\Delta x \end{cases}$$

$\varphi(x)$  = Value read directly from function table without interpolation

$Z(x)$  = last of the sequence of quantities  $Z_r(x)$

$Z_0(x) = Z(x - \Delta x)$  first approximation to  $Z(x)$

$$Z_r(x) = \frac{2Y(x)Z_{r-1}^3(x) + 1}{3Y(x)Z_{r-1}^2(x)} \quad r = 1, 2, 3, \dots$$

iterating until  $|Z_r - Z_{r-1}| \leq 1 \times 10^{-5} \quad 6 \times 10^{-6}$

$$Y(x) = 100 - a x$$

TABLE III - RANGES AND MAXIMUM ALLOWABLE ERRORS

All variables listed are monotonic functions of both  $x$  and  $a$

Variable	Value at $x = 0$	Value at $x = 10, a = 5$	Maximum Allowable Error
$x$	0	10	Exact
$Y(x)$	100	50	Exact
$Z(x)$	.216	.272	$6 \times 10^{-6}$
$\varphi(x)$	1.00	70.0	$5 \times 10^{-6}$ (given)
$f_a(x)$	.216	19.0	$1 \times 10^{-3}$
$F_a(x)$	0	147.0	$5 \times 10^{-3}$
$3Y^2$	14.1	11.1	$2 \times 10^{-4}$
$2YZ_r^3 + 1$	3	3	$5 \times 10^{-5}$

TABLE IV - INPUT QUANTITIES

$\bar{x}(0) = 1/\sqrt[3]{100}$

$a_0 = 0$

$x_0 = 0$

5 = upper limit of x

10 = upper limit of a

$\Delta a = .1$

$\Delta x = .1$

$T = 1 \times 10^{-5}$  tolerance for  $1/\sqrt[3]{Y}$

TABLE V - OUTPUT QUANTITIES

$a = 0.0$

$x$ xx.x	$\bar{F}_a(x)$ xxx.xxxx
1	
1	
1	
1	

$a = 0.1$

$x$ xx.x	$\bar{F}_a(x)$ xxx.xxxx
1	
1	
1	
1	

TABLE VI - SAMPLE RUN

$a = .3, x = 0$

$$f_a(0) = 1/\sqrt[3]{100} = .215\ 443$$

$a = .3, x = .1$

$$\varphi = 1.0025$$

$$Y = 99.97$$

$$r = 0$$

$$Z_r = .215\ 443$$

$$Z_r Y = 21.537\ 8$$

$$Z_r^2 Y = 4.640\ 17$$

$$Z_r^3 Y = .999\ 692$$

$$3Y Z_r^2 = 13.920\ 5$$

$$2Y Z_r^3 + 1 = 2.999\ 38$$

$$Z_{r+1} = .215\ 465$$

$$Z_{r+1} - Z_r = .000\ 022$$

$$.215\ 465$$

$$21.540\ 0$$

$$4.641\ 12$$

$$.999\ 999$$

$$13.923\ 4$$

$$2.999\ 99$$

$$.215\ 464$$

$$.000\ 001$$

$$f_a(.1) = \varphi \cdot Z = .216\ 003$$

$a = .3, x = .2$

$$\varphi = 1.0107$$

$$Y = 99.94$$

$$r = 0$$

$$Z_r = .215\ 464$$

$$Z_r Y = 21.533\ 5$$

$$Z_r^2 Y = 4.639\ 69$$

$$Z_r^3 Y = .999\ 686$$

$$3Y Z_r^2 = 13.919\ 1$$

$$2Y Z_r^3 + 1 = 2.999\ 37$$

$$Z_{r+1} = .215\ 486$$

$$Z_{r+1} - Z_r = .000\ 022$$

$$.215\ 486$$

$$21.535\ 7$$

$$4.640\ 64$$

$$.999\ 993$$

$$13.921\ 9$$

$$2.999\ 99$$

$$.215\ 487$$

$$.000\ 001$$

$$f_a(.2) = \varphi \cdot Z = .217\ 793$$

$$F_a(0) = 0$$

$$F_a(.1) = 5/12 \Delta x f_a(0) + 8/12 \Delta x f_a(.1) - 1/12 \Delta x f_a(.2)$$

$$= (.041\ 666\ 7)(.215\ 443) + (.066\ 666\ 7)(.216\ 003) - (.008\ 333\ 3)(.217\ 793)$$

$$= .021\ 562$$

$$F_a(.2) = 1/3 \Delta x f_a(0) + 4/3 \Delta x f_a(.1) + 1/3 \Delta x f_a(.2)$$

$$= (.033\ 333\ 3)(.215\ 443) + (.133\ 333)(.216\ 003) + (.033\ 333\ 3)(.217\ 793)$$

$$= .043\ 241$$

## X - THE DIFFERENTIAL ANALYZER

The Aberdeen Differential Analyzer resembles in its main characteristics the original analyzer invented by Bush, and constructed at M.I.T. in the years 1930-31. While neither quite as fast nor as accurate as the large scale computers discussed in the foregoing sections the machine is, nevertheless, very efficient and time-saving, especially in cases where the study of ordinary differential equations is called for.

The analyzer is based on principles entirely different from those governing the digital machines. It is an "analogue" machine, i.e., it duplicates physical situations by means of mechanical devices, in a manner to be described later. The machine will solve reasonably complicated differential equations of any order up to the tenth - theoretically at least but rarely in practice - within a relatively short time, to an accuracy of 3 - 4 digits. Its ruggedness, flexibility, and ease of handling have made it a valuable tool in the field of mathematical and physical research.

For detailed discussions of the analyzer the reader is advised to consult references (7) and (8) given in the Bibliography.

As already stated the differential analyzer solves mathematical problems by constructing their physical analogue as follows: to each value of every significant quantity occurring in the mathematical description there is assigned a position of a rotating shaft in the machine, so that changes in values are represented by clockwise or counter-clockwise rotations of shafts through corresponding angles. By means of proper linkages and inter-connections and the use of certain basic units, the shafts can be constrained to rotate in accordance with the relations prescribed in the mathematical formulation. The rotation of the shaft representing the independent variable - brought about by a motor - will then drive everything else.

The basic units referred to above, and most frequently employed, are input tables, integrators, adders, and output mechanisms. A known functional relationship may be introduced into the machine either analytically by generating it on the basis of subsidiary differential equations, such as, for example,  $y' = y$  for  $y = e^x$ , through the interconnection of appropriate analyzer components, or graphically, by means of a function table. Many analytic functions such as  $x^n$ ,  $e^x$ ,  $\sin x$ , etc., may be generated quite easily.

A function table has essentially two shafts with revolutions  $x$  and  $y$ , the first shaft moving a pointer horizontally in the direction of abscissas, the other one moving it vertically in the direction of ordinates across the plot  $y = f(x)$  spread out on the function table. Constant coefficients can be taken care of either by means of one of these function tables, or by the use of spur gears or trains of such gears.

The fundamental component of the analyzer is the integrator. It is essentially a planimeter, i.e. a device that produces a lower-order derivative from a higher-order one. It has two shafts for inputs  $u$  and

v, and one for output w, arranged such that the three quantities u, v, w are related to each other in the form  $dw = kuv$ , k being a certain constant. These integrators may be used to perform many different duties. Thus, e.g., multiplication may be accomplished by means of  $uv = \int u dv + \int v du$ . Similarly, the quotient  $u/v$  of two quantities is formed as the product  $u(1/v)$ , where the reciprocal  $1/v$  can be determined in a variety of ways.

The adder is actually a planetary differential gear. It can be connected to three shafts in such a manner that the number of revolutions of one equals the sum of the numbers of revolutions of the other two; algebraically the three quantities are related by  $u+v+w=0$ , if the senses of rotation are also taken into account.

Finally, the output of the analyzer may be obtained either graphically by means of the output table, a device very similar to the input table, or numerically, by using the recorder, a device which tabulates successive values of as many as six variables as functions of one of them, for equal intervals of that chosen variable.

The basic units just mentioned - and a few not described here, such as the combined polar input table and multiplier, and the vector table - suffice to carry out the basic operations of addition, subtraction, multiplication, division, integration, and generation of functions.

Let us now briefly describe the technique employed in setting-up the analyzer for the solution of an ordinary differential equation. Before going into detail it is perhaps worth-while stating that the method of solving problems on the analyzer, as on any of the large scale machines, is not a rigidly fixed one, but that a large variety of approaches in the treatment of many problems is possible.

Considering, then, such a problem as  $F(d^n y/dx^n, d^{n-1} y/dx^{n-1}, \dots, dy/dx, y, x) = 0$ , the first step usually consists in the solution of  $F=0$  for the highest order derivative  $d^n y/dx^n$  present:  $d^n y/dx^n = G(d^{n-1} y/dx^{n-1}, \dots, y, x)$ . However, the property of the equation  $F=0$  to permit convenient isolation of the highest order derivative present is not a necessary requirement for its tractability on the analyzer.

Next the expression  $G(d^{n-1} y/dx^{n-1}, \dots, dy/dx, y, x)$  must be broken down into basic operations of the type the machine components can perform. Finally, the individual components must be interconnected in such a manner that the grouping of terms and the relations among them are such as to satisfy the differential equation. All these interconnections must satisfy the rule of torques, which may be stated thus:

An analyzer assembly is operable when all the mathematical relations required by the equation have been established among the variables represented, and when every operating shaft is connected to one and only one source of torque.

In practice, the solution of a problem on the analyzer starts with the preparation of a connection diagram. In such a diagram the various

significant variables occurring in the problem are assigned individual shafts in the machine. Then the essential connections are established in accordance with the governing equation, accompanied by close observance of the rule of torques mentioned above. Finally, a second diagram is drawn, differing from the first one only in that scale factors and gear ratios are now introduced. A scale factor assigned to a shaft specifies the number of revolutions of that shaft which equals one unit of the quantity represented. When connections are established to integrators, input or output tables, etc., account must be taken of fixed scale factors given by the lead screws of these components. A lead screw is a shaft threaded so as to effectively constitute a long screw.

While scale factors are to some extent arbitrary and may be chosen conveniently, there are certain conditions resulting from the set-up, the various units employed, etc., that must be considered. Thus, when the final connections of the analyzer assembly are made there are obtained equalities which must be satisfied by the various scale factors. Thus, for example, the scale factor of the output quantity of an adder must be equal to the scale factors of the inputs. If, then, one of the input quantities of such an adder is, say,  $(n_4C) \sin y$ , the other input quantity is  $(n_2B)y^1$ , then necessarily  $n_4C = n_2B$ . Further, displacements of integrators and function tables must be kept within the limits of their design. The lead screw of an integrator, for example, admits only 40 turns. In this manner there are then introduced certain inequalities for the scale factors.

In addition to these, certain other considerations also affect the final choice of the factors. It is obvious that a quantity is determined with greater precision if the number of revolutions of its shaft that is to equal unity is made larger. On the other hand one desires the time required for a solution to be reasonably low.

When the factors have thus been determined a final diagram is made, scale factors and gear ratios are indicated, the direction of rotation of each shaft is shown, plots are made for the functions to be placed on the function tables, initial values are set on the lead screws of function units, integrators, printing counters, etc., and the machine is about ready to operate. The time required to work out a differential analyzer assembly that will solve a given problem depends, naturally, on the type of problem considered. A simple equation may be set up within a few hours. A more complicated one, however, especially if it is different from most of the equations dealt with previously, may take several days' work to be set up in a reasonably good way.

Working out a good assembly is important not only for the solution of the equation as a whole but also because it has some effect on the time required for the running off of each particular solution of the problem. Varying between a few minutes to more than an hour, depending on the nature and range of the solution itself, the running time quite obviously becomes important especially in cases where large numbers of solutions are desired.

On the basis of the foregoing description it is apparent what information must be required of a problem that is to go on the analyzer. This information is about of the same kind as that expected of the digital machine. Enough initial data must be provided to fix the solution. For the proper assignment of scale factors, consistent with accuracy requirements, it is necessary to know, at least approximately, the ranges of all the variable quantities involved. Finally, it is very desirable to have on hand a pre-computed typical case in order to check first results obtained on the machine.

A final word on the types of problems best suited for the differential analyzer. Differential equations requiring many multiplications or divisions - operations that tie up many integrators - may frequently not be suitable, especially if their order is relatively high. Problems where only few solutions are defined are usually not suited for the machine since in such cases the time required to work out a set-up may be out of proportion as compared with the actual running time. Further, solutions where high accuracy is demanded may be impossible of achievement.

While each particular problem must be examined individually, the remarks just made do point out certain general characteristics that should be kept in mind.

During the last war the differential analyzer was the only large scale computing device available at the Ballistic Research Laboratories. It was then used almost exclusively for the computation of trajectories that go into the making of firing tables. Since the end of the war, the machine has been successfully put to work on a wide variety of problems arising in the field of ballistic research.

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
6	Chief of Ordnance Washington 25, D. C. ATTN: ORDTB-Bal Sec	1	Commanding Officer Watertown Arsenal Watertown 72, Mass.
5	British - to ORDTB for distribution (Of interest to: Prof. D. R. Hartree, Cambridge University)	1	Watson Scientific Computing Laboratory 612 W. 116th St. New York 27, N. Y.
2	Chief, Bureau of Ordnance Navy Department Washington 25, D. C. ATTN: Re3	1	Computation Laboratory Harvard University Cambridge 38, Mass.
2	Chief, Bureau of Aeronautics Navy Department Washington 25, D. C.	1	Commanding Officer Naval Proving Ground Dahlgren, Virginia
2	Office of Naval Research Navy Department Math. Branch, Physical Science Division Washington 25, D. C. ATTN: Dr. Mina Rees	1	Langley Memorial Aeronautical Laboratory Langley Field Hampton, Virginia
2	Commanding General Air Materiel Command Wright-Patterson Air Force Base Wright Field, Dayton, Ohio ATTN: MCREON	1	Computation Laboratory National Applied Math. Laboratories National Bureau of Standards Washington 25, D. C.
2	Atomic Energy Commission Washington 25, D. C.	1	Institute for Numerical Analysis National Bureau of Standards 405 Hilgard Ave. Los Angeles 24, Calif.
1	Commanding Officer Picatinny Arsenal Dover, N. J.	1	Mass. Inst. of Tech. Cambridge 39, Mass.
1	Commanding General Frankford Arsenal Philadelphia 37, Pa.	1	Moore School of Electrical Engr. Univ. of Pennsylvania Philadelphia, Pa.